



A34941 - 066340.0141

PATENT

0300  
#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Wolfram Drescher  
Serial No. : 10/071,708  
Filed : February 8, 2002  
For : PROCESS AND APPARATUS FOR FINITE FIELD  
MULTIPLICATION (FFM)

I hereby certify that this paper is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231, on:

March 26, 2002

Date of Deposit

James J. Maune

Attorney Name

26,946

PTO Reg. No.

March 26, 2002

Date of Signature

Signature

CLAIM FOR PRIORITY UNDER 35 U.S.C. §119

Assistant Commissioner of Patents

Washington, D.C. 20231

Sir:

A claim for priority is hereby made under the provisions of 35 U.S.C. §119 for the above-identified U.S. patent application based upon German patent Application No. 101 06 085.8-53 filed February 8, 2001. A certified copy of this application is enclosed.

Respectfully submitted,

James J. Maune  
Patent Office Reg. No. 26,946  
Attorney for Applicant  
212-408-2566

Baker Botts LL.P  
30 Rockefeller Plaza  
New York NY 10112

NY02:377690.1

# BUNDESREPUBLIK DEUTSCHLAND



## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 101 06 085.8  
**Anmeldetag:** 08. Februar 2001  
**Anmelder/Inhaber:** Systemonic AG, Dresden/DE  
**Bezeichnung:** Verfahren und Anordnung zur Finiten  
Feld Multiplikation  
**IPC:** G 06 F 7/72

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 07. Februar 2002  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

Agurks

**LIPPERT, STACHOW, SCHMIDT & PARTNER**

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51.3 18 18-0

Telefax +49 (0) 3 51.3 18 18 33

Ad-wb/wb

8. Februar 2001

5      **Systemonic AG**  
       01099 Dresden

Verfahren und Anordnung zur Finiten Feld Multiplikation  
 (FFM)

Die Erfindung betrifft ein Verfahren zur Ausführung einer  
 Finiten Feld Multiplikation eines in einem ersten Eingangs-  
 register eines mit den Bitstellen  $a_0 \dots a_{n-1}$  bereitgestellten  
 ersten Galoiselementes  $a$ , welches mit einem zweiten, in einem  
 zweiten Eingangsregister mit den Bitstellen  $b_0 \dots b_{n-1}$  be-  
 reitgestellten Galoiselementes  $b$  in einem digitalen Galois-  
 Multiplizierer (MUL) multipliziert wird, wobei die Galois-  
 elemente  $a$  und  $b$  zu einem Galoisfeld  $GF\ 2^n$ , welches durch ein  
 irreduzibles Polynom  $PR$  mit den Bitstellen  $PR_0 \dots PR_{n-1}$   
 beschrieben wird, gehören.

Es wird bei der Finiten Feld Multiplikation von Galoiselemen-  
 ten darauf Bezug genommen, dass die kleinste Grundmenge von  
 arithmetischen Operationen in der digitalen Signalverarbeitung  
 allgemein mit

$$y = \sum_{i=0}^n x_i * a_i$$

angegeben wird. Viele Algorithmen lassen sich im Kern auf  
 diese Faltungssumme, oder arithmetisch betrachtet, auf die  
 Summenbildung über Produkte, zurückführen. Üblicherweise wer-  
 den in der digitalen Signalverarbeitung diese Algorithmen z.B.  
 in digitalen Signalprozessoren (DSP) durch die Realisierung  
 einer fest verdrahteten Hardware-Baugruppe, die diese Produkt-  
 summe implementiert, beschleunigt. Solche Baugruppen werden  
 allgemein als Multiplizierer (MUL) bezeichnet.

Wird bei diesem Multiplizierer die Arithmetik in Restklassen-  
 körpern mit ihrer Modulo-Operation angewandt, kommt ein  
 Galois-MUL zur Anwendung, welcher eine finite Feldmultiplika-  
 tion ausführt, indem in jeder Zeile des Multiplizierers zuerst  
 5 ein partielles Produkt  $a * b_i$  ( $i = 0$  bis  $n-1$ ) gebildet wird.  
 Danach wird das partielle Produkt mit der Summe aus den vorher-  
 rigen Zeilen summiert bevor die Modulo-Operation ausgeführt  
 wird. Dies geschieht, indem in Abhängigkeit von einem aufge-  
 tretenen Überlauf in der vorherigen Zeile die Bitstellen  $PR_0$   
 10 ... $PR_{n-1}$  zur vorher berechneten Summen addiert werden.

Die Erfindung betrifft weiterhin je eine Schaltungsanordnung  
 zur Durchführung des oben genannten Verfahrens nach den An-  
 sprüchen 1 bis 3 und eine Schaltungsanordnung bei der ein  
 15 Galois-Multiplizierer-Accumulator (MAC) angeordnet ist, in  
 welchem diese genannten Schaltungsanordnungen jeweils enthal-  
 ten sind.

Ein zellular aufgebauter Galois-MUL wurde in "A Cellular-  
 20 Array-Multiplier for  $GF(2^m)$ ", IEEE Transactions on Computers,  
 Dez.1971, S.1573-1578 von B.A.Laws, C.K.Rushforth beschrieben.  
 In "Efficient Semisystolic Architectures for Finite-Field  
 Arithmetic", IEEE Transactions on very large scale integration  
 (VLSI) systems, Vol.6 Nr 1, März 1998, S.101-113 werden von  
 25 Surendra K.Jain, Leilei Song und Keshab K.Parhi verschiedene  
 Algorithmen für die Realisierung von Galois-Multiplizierern  
 und deren schaltungstechnische Umsetzung gegenübergestellt.

Nachteilig an den bisher angegebenen Realisierungen ist, daß  
 30 sie eine hohe "logische Tiefe", d.h. eine Vielzahl hinterein-  
 ander zu durchlaufende Gatter, aufweisen und deshalb bei ihrer  
 Implementierung lange Signallaufzeiten bei der finiten Feld-  
 multiplikation auftreten.

35 Der Erfindung liegt die Aufgabe zugrunde, die finite Feldmul-  
 tiplikation zu beschleunigen.

Die verfahrenseitige Lösung der Aufgabenstellung sieht vor,

dass in einem ersten Teil des Galois-MUL, einem Addierteil, in einem Verarbeitungsschritt ein Zwischenergebnis  $Z$  von Zwischensummen partieller Produkte der Bitbreite  $2n-2$  gebildet wird, welches kein Element des durch ein irreduzibles Polynom  $PR$  beschriebenen Galoisfeldes repräsentiert, und dass in einem zweiten Teil des Galois-MUL, einem Reduzierteil, das Zwischenergebnis  $Z$  mit den Bitstellen  $Z_{2n-2} \dots Z_0$  mittels einer Modulo-Division durch das irreduzible Polynom  $PR$  mit den Bitstellen  $PR_{n-1} \dots PR_0$  verarbeitet wird und damit nach Durchlaufen aller XOR-Verknüpfungen das Ergebnis  $E$  mit den Bitstellen  $E_{n-1} \dots E_0$  berechnet wird.

Bei dieser erfindungsgemäßen Lösung wird im Addier-Teil eine Baumstruktur realisiert, die wegen ihrer parallelen Signalverarbeitung beschleunigter als beim Stand der Technik arbeitet.

Eine vorteilhafte Variante der verfahrenseitigen Lösung sieht vor, dass in einem Reduzierteil die Modulo-Division des Zwischenergebnisses  $Z$  durch das irreduzible Polynom  $PR$  zweistufig ausgeführt wird, indem in einer ersten Verarbeitungsstufe alle Bitstellen  $Z_{2n-2} \dots Z_n$  mit einer erweiterten Form  $PE$  des irreduziblen Polynoms  $PR$  mit den Bitstellen  $PE_{n-1} \dots PE_0$  jeweils AND-verknüpft werden und danach mittels einer ersten parallel operierenden Adder-Baumstruktur, welche die logische Operation XOR ausführend realisiert, zusammengefaßt werden.

Diese zusammengefaßten Teilergebnisse werden nachfolgend in einer zweiten Verarbeitungsstufe mit den Bitstellen  $PR_{n-1} \dots PR_0$  des irreduziblen Polynoms  $PR$  jeweils AND-verknüpft und sie werden mittels einer zweiten parallel operierenden Adder-Baumstruktur, welche auch die logische Operation XOR mehrfach ausführend realisiert, jeweils zusammen mit den Bitstellen  $Z_{n-1} \dots Z_0$  des Zwischenergebnisses  $Z$  zum Ergebnis  $E$  mit den Bitstellen  $E_{n-1} \dots E_0$  zusammenfaßt.

Bei dieser erfindungsgemäßen Lösung wird realisiert, daß auch im Reduzierteil durch eine zweistufige Parallelführung der Signalwege zusätzlich die Signallaufzeiten vermindert werden.

Eine sehr vorteilhafte erfindungsgemäße Lösung der Aufgabenstellung sieht vor, dass eine Matrix PEM der Bitstellenanordnung

$$5 \quad PEM = \begin{pmatrix} PE_{n-1,n-2} & \dots & PE_{n-1,0} \\ \vdots & & \vdots \\ PE_{0,n-2} & \dots & PE_{0,0} \end{pmatrix}$$

10 bezüglich der erweiterten Form PE des irreduziblen Polynoms PR im Galois-Feld  $GF\ 2^n$  vorausberechnet wird, dass im Reduzierteil die Modulo-Division des Zwischenergebnisses Z durch das irreduzible Polynom PR dadurch ausgeführt wird, dass alle Bits  $Z_{2n-2} \dots Z_n$  mit der Matrix PEM der erweiterten Form PE des irreduziblen Polynoms PR mit den Bitstellen  $PE_{n-1,n-2} \dots PE_{n-1,0} \dots$   
 15  $PE_{0,n-2} \dots PE_{0,0}$  jeweils AND-verknüpft werden und danach mittels einer dritten parallel operierenden Adder-Baumstruktur, die die logische Operation XOR realisiert, zum Ergebnis E zusammengefaßt werden.

20 Bei dieser erfindungsgemäßen Lösung wird eine weitere Laufzeitverkürzung bewirkt, in dem alle Bitstellen  $Z_{2n-2} \dots Z_n$  des Zwischenergebnisses Z durch einen vorausberechneten Datensatz von erweiterten Formen PEM des irreduziblen Polynoms, welche matrixartig im Reduzierteil des Galois-MUL verarbeitet werden,  
 25 AND-verknüpft werden und nachfolgend in einer einstufigen dritten Parallel-Adder Baumstruktur zusammengefaßt werden.

Diese Anwendung des vorberechneten erweiterten irreduziblen Polynoms bei der Modulo-Operation wirkt sich durch die einstufige Verarbeitung im Reduzierteil besonders beschleunigend  
 30 für die Ausführung der Finiten Feldmultiplikation aus.

Eine vorteilhafte Ausführung der erfindungsgemäßen Lösung der Aufgabenstellung sieht vor, dass bei variablen Galoiselementen  
 35  $va < a$  und  $vb < b$ , mit einer hierbei resultierenden Bitbreite  $2m-2$  des Zwischenergebnisses Z zur Anpassung an unterschiedliche Galois-Felder, alle Bitstellen des Zwischenergebnisses Z bevor sie mit der erweiterten Form PE des irreduziblen Polynoms PR

verknüpft werden, um  $Z_{2^{m_{\max}}-2}-Z_m$  Stellen mittels eines Decoders und Feldgrößenadaptionslogik verschoben werden, wobei die Bitbreite  $2^{m_{\max}}$  die maximale Bitbreite des Zwischenergebnisses  $Z$  bei jeweils maximaler Bitbreite  $n$  der Galoiselemente  $a$  und  $b$  darstellt.

Bei dieser Lösung wird berücksichtigt, dass der Galois-MUL programmierbar und damit an die Größe seiner Verarbeitungsaufgaben anpassbar gestaltet wird. Dies wird dadurch erreicht, dass führende Nullen bei den Verarbeitungsoperationen durch Linksverschiebung der Operanden bei der Eingabe in den Galois-MUL nicht berücksichtigt werden.

Eine anordnungsseitige vorteilhafte Ausführung der erfindungsgemäßen Lösung der Aufgabenstellung sieht vor, dass ein Galois-Multiplizierer-Accumulator (MAC) angeordnet ist, der den Galois-MUL, das erste und zweite Eingaberegister, die erweiterte Form  $PE$  und/oder das irreduzible Polynom  $PR$  oder die die Matrix  $PEM$  enthaltende Preset-Register-Bank, das Ergebnisregister und einen Addierer enthält, wobei der Ausgang des Galois-MUL mit einem Eingang des Addierers verbunden ist. Dessen Ausgang wiederum ist mit einem Eingang des ersten Eingaberegisters und/oder einem Eingang des zweiten Eingaberegisters geschaltet.

In dieser anordnungsseitigen Lösung wird die vorteilhafte Anwendung des erfindungsgemäßen Galois-MUL innerhalb eines Galois-Multiplizierer-Accumulator (MAC) komplex ausgestaltet.

Die Erfindung soll nachfolgend anhand eines Ausführungsbeispiels näher erläutert werden. In den zugehörigen Zeichnungen zeigt:

Fig. 1 ein Blockschaltbild eines Galois-MUL mit Bitmultiplizierer und Partiell-Produkt-Adder im Addierteil und einem Reduzierteil mit Reduktions-Modulo-Anordnung,

Fig. 2 ein Blockschaltbild eines Galois-MUL mit Bitmultipli-

zierer und Partiell-Produkt-Adder im Addierteil und im Reduzierteil eine Reduktions-Zweistufen-Adder-Anordnung, die Q-Reduktions-Adderteil und Final-Adderteil realisiert,

5

Fig. 3 ein Blockschaltbild eines Galois-MUL mit Bitmultiplizierer und Partiell-Produkt-Adder im Addierteil und im Reduzierteil mit Reduktions-Einstufen-Adder-Anordnung,

10

Fig. 4 ein Blockschaltbild eines Galois-MUL mit Bitmultiplizierer und Partiell-Produkt-Adder im Addierteil und Reduzierteil mit Decoder und Feldgrößenadaptionslogik

15

Fig. 5 ein Schaltbild einer Partiell-Produktadder-Schlüsselkonfiguration,

Fig. 6 ein Schaltbild einer Reduzierer-Modulo-Schlüsselkonfiguration,

20

Fig. 7 ein Schaltbild einer Zweistufen-Adder-Schlüsselkonfiguration und

25

Fig. 8 ein Schaltbild einer Einstufen-Adder-Schlüsselkonfiguration

30

Wie in Fig. 1 ersichtlich, wird im ersten Eingabe-Register 1 ein gespeichertes Galoiselement a und ein im zweiten Eingabe-Register 2 gespeichertes Galoiselement b für den Galois-MUL 20 bereitgestellt. Die Eingabe erfolgt in den Addierteil 105 des Galois-MUL 20 der weiterhin ein Reduzierteil 106 aufweist. Dieser Reduzierteil 106 ist als Reduktions-Modulo-Anordnung 107 ausgeführt. Innerhalb des Galois-MUL 20 erzeugt der Addierteil 105 ein Zwischenergebnis zur Modulo-Verarbeitung in der Reduktions-Modulo-Anordnung 107. Weiterhin wird am Eingang der Reduktions-Modulo-Anordnung 107 das gespeicherte irreduzible Polynom PR 9 bereitgestellt.

35



Der Addierteil 105 besteht aus einem Bitmultiplizierer 3 und aus einem Partiell-Produkt-Adder 4. Die in den Galois-MUL 20 eingegebenen Galoiselemente a und b werden bezüglich ihrer Stellenwerte einzeln im Bitmultiplizierer 3 ausmultipliziert. Diese werden als einzelne partielle Produkte für den Partiell-Produkt-Adder 4 zur Verfügung gestellt und werden in diesem zu den einzelnen Stellenwerten  $Z_{2n-2} \dots Z_0$  addierend zusammengefaßt. Der Addierteil 105 besteht in seiner Anordnung aus der Partiell-Produktadder-Erweiterungsstruktur 16 und der Partiell-Produktadder-Schlüsselkonfiguration 19.

Das Zwischenergebnis Z wird mit dem Wert des gespeicherten irreduziblen Polynoms PR 9 in der Reduktions-Modulo-Anordnung 107 modulo-verknüpft und kann hier am Ausgang mit dem Wert E für das angeschlossene Ergebnisregister 10 bereitgestellt werden.

Die Reduktions-Modulo-Anordnung 107 besteht in ihrer Anordnung aus einer Reduzierer-Modulo-Erweiterungsstruktur 17 und der Reduzierer-Modulo-Schlüsselkonfiguration 18.

Wie in Fig. 2 ersichtlich, sind der Aufbau und die Wirkungsweise des Addierteiles 105 des hier beschriebenen Galois-MUL übereinstimmend mit dem in Fig. 1 beschriebenen. Abweichend davon wird hierbei zur Ausführung der Modulo-Verknüpfung im Reduzierteil 106 zusätzlich eine vorausberechnete erweiterte Form PE 7 des irreduziblen Polynoms PR 9 angeschlossen.

Diese erweiterte Form PE 7 des irreduziblen Polynoms PR 9 wird in drei Teilschritten berechnet, wobei die Vereinbarung gilt:

- $p(0)$  bezeichnet das MSB und  $p(n)$  das LSB des irreduziblen Polynoms,
- $n$  ist eine Feldgröße

Es werden die Terme mit der Matrix

$$Z^{n \times n} = \begin{pmatrix} Z_{11} & \dots & Z_{1n} \\ \vdots & & \vdots \\ Z_{n1} & \dots & Z_{nn} \end{pmatrix}$$

berechnet und der folgende Berechnungsablauf angewandt:

1. initialisieren der Matrix  $Z_{ij}$ : = 1 alle anderen Elemente  
der 1. Zeile auf Null setzen  $Z_{1j}$ : = 0, mit  $j = 2 \dots n$
2. Berechnung der Zeilen  $i = 2 \dots n$  mit der rekursiven Formel:  
 $Z_{ij} = Z_{(i-1), (j+1)} \text{ XOR } (p(j) \text{ AND } Z_{(i-1), 1})$  mit  $i = 2 \dots n, j = 1 \dots n-i+1$
3. Ablesen der Preset-Terme:  $r(i-1) = Z_{i,1}$  mit  $i = 1 \dots n$  und  
Einspeichern in das Preset-Register 7.

Der das Zwischenergebnis Z verarbeitende Reduzierteil 106 ist hierbei zweistufig, aus Q-Reduktionsadderteil 5 und Final-Adderteil 6 bestehend, gestaltet.

Im Q-Reduktionsadderteil 5 werden alle Stellenwerte  $Z_{2n-2} \dots Z_n$  des an seinem Eingang anliegenden Zwischenergebnisses Z jeweils mit den Bitstellen des erweiterten Polynoms AND-verknüpft, zeilenweise in Q-Reduktionsaddern modulo-addiert und diese Werte für die Weiterverarbeitung im nachfolgend angeschlossenen Final-Adderteil 6 bereitgestellt. Hier werden sie jeweils mit den Bitstellen des am Eingang des Final-Adderteils 6 angeschlossenen irreduziblen Polynoms PR 9 AND-verknüpft und anschließend werden diese Werte spaltenweise zusammen mit den Stellenwerten  $Z_{n-1} \dots Z_0$  des Zwischenergebnisses Z im Final-Adderteil 6 modulo-addierend zusammengefaßt und am Ausgang des Final-Adderteiles 6 an das Ergebnisregister 10 ausgegeben.

Die Reduktions-Zweistufen-Adder-Anordnung 110 besteht in ihrer Anordnung aus der Zweistufen-Adder-Schlüsselkonfiguration 109 und der Zweistufen-Adder-Erweiterungsstruktur 108.

Wie in Fig. 3 ersichtlich, sind der Aufbau und Wirkungsweise des Addierteiles 105 des hier beschriebenen Galois-MUL übereinstimmend mit dem in Fig. 1 und Fig. 2 beschriebenen. Abweichend davon wird hierbei zur Ausführung der Modulo-Verknüpfung im Reduzierteil 106 an Stelle von irreduziblen Polynom PR 9, bzw. erweiterte Form PE 7, eine Preset-Register-Bank 12, in welcher eine Matrix PEM 11 vorausberechneter Terme des

irreduziblen Polynoms gespeichert wird, zum Einsatz gebracht.

Für die Berechnung der Preset-Terme, die bei dieser Reduktions-Einstufen-Adder-Anordnung 13 zur Anwendung gelangen, gilt die Vereinbarung:

- $P_0$  bezeichnet das LSB des irreduziblen Polynoms PR 9
- $PR_{ij}$  bezeichnet j-ten Preset-Term zur Berechnung des i-ten Ergebnisses
- $n$  bezeichnet Feldgröße

und der folgende Berechnungsablauf angewandt:

$$PR_{ij} = (r_{j-1} \text{ AND } P_i \text{ XOR } r_{j-2} \text{ AND } P_{i-1} \text{ XOR } \dots \text{ XOR } r_{j-i-1} \text{ AND } P_0)$$

Wenn ein Index der r-Terme oder P-Terme  $< 0$  wird, so wird die Berechnung abgebrochen.

Der das Zwischenergebnis Z verarbeitende Reduzierteil 106 ist ist durch die Reduktions-Einstufen-Adder-Anordnung 113 realisiert. In ihm werden alle Stellenwerte  $Z_{2n-2} \dots Z_n$  des an seinem Eingang anliegenden Zwischenergebnisses Z jeweils mit der Matrix der Bitstellen der angeschlossenen Preset-Register-Bank 12 AND-verknüpft, und anschließend werden diese Werte zeilenweise zusammen mit den jeweiligen Stellenwerten  $Z_{n-1} \dots Z_0$  des Zwischenergebnisses Z in Addern modulo-addierend zusammengefaßt und an den Ausgängen der Adder wird das Ergebnis an das Ergebnisregister 10 bereitgestellt.

Die Reduktions-Einstufen-Adder-Anordnung 113 besteht in ihrer Anordnung aus der Einstufen-Adder-Schlüsselkonfiguration 112 und der Einstufen-Adder-Erweiterungsstruktur 111.

Wie in Fig. 4 ersichtlich, unterscheidet sich der Aufbau und die Wirkungsweise des hier beschriebenen Galois-MUL 20 von den in Fig. 1 und Fig. 2 beschriebenen darin, dass zusätzlich ein Decoder 14 und eine Feldgrößenadaptionlogik 13 bei der Verarbeitung des Zwischenergebnisses Z im Reduzierteil 106 zum Einsatz gelangt.

Je nach Bitbreite der im ersten Eingaberegister 1 und im zweiten Eingaberegister 2 gespeicherten Galoiselemente werden nach Verarbeitung im Addierteil 105 bei auftretenden führenden Nullen im Zwischenergebnis Z die Bitstellenbelegungen von Z so weit nach links verschoben, bis keine führenden Nullen bei der Weiterverarbeitung des Zwischenergebnisses anfallen. In der Feldgrößenadaptionslogik 13 ist diese Schieberegisterfunktion realisiert.

Wie in Fig. 5 ersichtlich, werden die an dem Bitstellenanschluss  $a_{n-1}$  23, Bitstellenanschluss  $a_{n-2}$  24, Bitstellenanschluss  $b_{n-1}$  25 und dem Bitstellenanschluss  $b_{n-2}$  26 anliegenden Stellenwerte der Galoiselemente a und b in den ersten bis sechsten XOR-Bitstellenmultiplizierergliedern 28, 30, 32, 34, 36, 38 ausmultipliziert und nachfolgend von den ersten bis sechsten XOR-Addergliedern 27, 29, 31, 33, 35, 37 modulo-addierend stellenweise zusammengefasst und an den ersten bis vierten Zwischenergebnisanschlüssen 39, 40, 41, und 42 zur Weiterverarbeitung an den Reduzierteil 106 bereitgestellt.

Die ersten und zweiten sowie fünften und sechsten XOR-Adderglieder 27, 29 bzw. 35, 37 sind jeweils mit einem Eingang mit den ersten bis vierten Partiell-Produktadder-Erweiterungsanschlüssen 21, 22 bzw. 43, 44 verbunden. Über diese Erweiterungsanschlüsse sind diese zusammenfassenden XOR-Adderglieder mit weiteren XOR-Addergliedern der Partiell-Produktadder-Erweiterungsstruktur 16 verbunden, in denen weitere partielle Produkte von weiteren Bitstellenwerten der eingespeicherten Galoiselementen a und b zusammengefasst werden, die mit ihren Produkt-Anteilen in die an den ersten bis vierten Zwischenergebnisanschlüssen 39, 40, 41, und 42 anliegenden Stellenwerte des Zwischenergebnisses mit eingehen.

Wie in Fig. 6 ersichtlich, realisiert die Anordnung den Algorithmus eines MSB-First Galois Multiplizierers und es gelangen über die ersten bis vierten Reduzierer-Zwischenergebnisanschlüsse 49 bis 52 die von der Reduzierer-Modulo-Erweiterungsstruktur 17 bereitgestellten Stellenwerte eines bearbeiteten

Zwischenergebnisses in die Reduzierer-Modulo-Schlüsselkonfiguration 18, welche die Signalverarbeitung in zwei vertikalen Verarbeitungsebenen vornimmt.

5 Außerdem werden über die Anschlüsse des ersten irreduziblen-Polynom-Registeranschlusses 45 und des zweiten irreduziblen-Polynom-Registeranschlusses 46 die Bitstellen  $PR_{n-1}$  und  $PR_{n-2}$  des gespeicherten irreduziblen Polynoms PR 9 bereitgestellt. Diese liegen somit am ersten AND-Gatter 53 und dritten AND-Gatter 57 bzw. am zweiten AND-Gatter 54 und vierten AND-Gatter 10 58 an und werden mit den jeweiligen bearbeiteten MSB-Stellenwert des Zwischenergebnisses Z getort.

Diese bearbeiteten Stellenwerte des Zwischenergebnisses Z 15 bilden gleichzeitig die MSB-Torsignale der jeweiligen Verarbeitungsebene und somit die MSB-Torsignale für die AND-Verknüpfung weiterer Bitstellen des irreduziblen Polynoms PR 9. Daher liegen die MSB-Torsignale einerseits für die erste Verarbeitungsebene auch an einem Eingang des zweiten AND-Gatters 20 54 und zur Weiterverarbeitung in der Reduzierer-Modulo-Erweiterungsstruktur 17 am zweiten Reduzierer-Erweiterungsanschluss 62 und andererseits für die zweite Verarbeitungsebene an einem Eingang des vierten AND-Gatters 58 an und zur Weiterverarbeitung in der Reduzierer-Modulo-Erweiterungsstruktur 17 am 25 ersten Reduzierer-Erweiterungsanschluss 61 an.

Für die Bitstelle  $PR_{n-1}$  des gespeicherten irreduziblen Polynoms PR 9 repräsentiert das MSB-Torsignal in der ersten Verarbeitungsebene die Signalbelegung des ersten Reduzierer-Zwischenergebnisanschlusses 49 und in der zweiten Verarbeitungsebene 30 die vom ersten XOR-Verknüpfers 55 bereitgestellten modulo-Verknüpfung der Signalbelegung vom zweiten Reduzierer-Zwischenergebnisanschluß 50 mit dem Ausgangssignal des erste AND-Gatters 53.

35 Weiterhin wird die Signalbelegung des bearbeiteten Stellenwerte des Zwischenergebnisses Z, welches am dritten Reduzierer-Zwischenergebnisanschluß 51 anliegt, mit dem Aus-

gangssignal des zweiten AND-Gatters 54, welches die AND-Verknüpfung des MSB-Torsignales der ersten Verarbeitungsebene mit der Bitstelle  $PR_{n-2}$  des irreduziblen Polynoms PR 9 bereitstellt, im zweiten XOR-Verknüpfen 56 modulo-multipliziert.

5 Dessen Ausgangssignal wird zusammen mit dem Ausgangssignal des dritten AND-Gatters 57, welches die AND-Verknüpfung des MSB-Torsignales der zweiten Verarbeitungsebene mit der Bitstelle  $PR_{n-1}$  des irreduziblen Polynoms PR 9 vornimmt, an den Eingängen des dritten XOR-Verknüpfers 59 zur weiteren modulo-Multiplikation angelegt. Wiederum stellt dessen Ausgangssignal die Signalbelegung für den Ergebnisanschluss  $E_{n-1}$  47 bereit.

Weiterhin wird die Signalbelegung des bearbeiteten Stellenwertes des Zwischenergebnisses Z, welches am vierten Reduzierer-Zwischenergebnisanschluss 52 anliegt, mit dem Ausgangssignal des vierten AND-Gatters 58, welches die AND-Verknüpfung des MSB-Torsignales der zweiten Verarbeitungsebene mit der Bitstelle  $PR_{n-2}$  des irreduziblen Polynoms PR 9 bereitstellt, im vierten XOR-Verknüpfen 60 modulo-multipliziert. Dessen Ausgangssignal stellt die Signalbelegung für den Ergebnisanschluss  $E_{n-2}$  48 bereit.

Wie in Fig. 7 ersichtlich, realisiert die Zweistufen-Adder-Schlüsselkonfiguration 109 die Signalverarbeitung in zwei vertikalen Ebenen in zwei horizontal aufgeteilten Adder-Stufen. Es gelangen über den ersten bis vierten Zwischenergebnisanschluss 39 bis 42 die von der Zweistufen-Adder-Erweiterungsstruktur 108 bereitgestellten Stellenwerte des Zwischenergebnisses Z in die Zweistufen-Adder-Schlüsselkonfiguration 109.

Außerdem gelangen über die Anschlüsse des ersten Irreduzibel-Polynom-Registeranschlusses 45 und zweiten Irreduzibel-Polynom-Registeranschlusses 46 die Bitstellen  $PR_{n-1}$  und  $PR_{n-2}$  des gespeicherten irreduziblen Polynoms PR 9 sowie über den ersten Preset-Registeranschluss 82 und den zweiten Preset-Registeranschluss 85 die Bitstellen  $PE_{n-1}$  und  $PE_{n-2}$  der gespeicherten erweiterten Form PE 7 des irreduziblen Polynoms PR 9 in die Zweistufen-Adder-Schlüsselkonfiguration 109.

Die über den ersten Zwischenergebnisanschluss 39 bereitgestellte Bitstelle  $Z_{n+1}$  des Zwischenergebnisses  $Z$  wird einerseits am Eingang des siebenten Zellgatters 94 mit der Bitstelle  $PE_{n-1}$  der erweiterten Form PE 7 des irreduziblen Polynoms PR 9 und andererseits am Eingang des fünften Zellgatters 91 mit der Bitstelle  $PE_{n-2}$  der erweiterten Form PE 7 des irreduziblen Polynoms PR 9 AND-verknüpft. Die über den zweiten Zwischenergebnisanschluss 40 bereitgestellte Bitstelle  $Z_n$  des Zwischenergebnisses  $Z$  wird am Eingang des achten Zellgatters 95 ebenfalls mit der Bitstelle  $PE_{n-2}$  der erweiterten Form PE 7 des irreduziblen Polynoms PR 9 AND-verknüpft.

Die Ausgänge des siebenten und achten Zellgatters 94, 95 werden im ersten Erweiterungsgatter 96 zusammengefaßt und nachfolgend wird das an dessen Ausgang anliegende Summensignal mit einem weiteren Summensignal der zweiten Verarbeitungsebene der Zweistufen-Adder-Erweiterungsstruktur 108, welche über den fünften-Reduzier-Erweiterungs-Anschluß 81 bereitgestellt wird, im zweiten Erweiterungsadder 98, ebenfalls modulo-addiert.

Das Ausgangssignal des fünften Zellgatters 91 wird mit einem weiteren Summensignal der ersten Verarbeitungsebene der Zweistufen-Adder-Erweiterungsstruktur 108, welche über den sechsten-Reduzier-Erweiterungs-Anschluß 84 bereitgestellt wird, am Eingang des dritten Erweiterungsadders 92 modulo-addiert. Das Ausgangssignal des dritten Erweiterungsadders 92 bildet das MSB-Torsignal der ersten Verarbeitungsebene und wird dazu einerseits an das sechste Zellgatter 93 geschaltet und andererseits an den neunten Reduzier-Erweiterungs-Anschluss 86 angelegt, um eine weitere Verarbeitung in der ersten Verarbeitungsebene der Zweistufen-Adder-Erweiterungsstruktur 108 zu gewährleisten.

Der Ausgang des zweiten Erweiterungsadders 98 bildet das MSB-Torsignal der zweiten Verarbeitungsebene und wird dazu einerseits an die Eingänge des neunten Zellgatters 97 und des zehnten Zellgatters 101 geschaltet und andererseits an den zehnten Reduzier-Erweiterungs-Anschluss 89 angelegt, um eine weitere

Verarbeitung in der zweiten Verarbeitungsebene der Zweistufen-Adder-Erweiterungsstruktur 108 zu gewährleisten.

Das Ausgangssignal des dritten Erweiterungsadders 92 wird mit der Bitstelle  $PR_{n-2}$  des gespeicherten irreduziblen Polynoms PR 9, welche über den zweiten irreduziblen-Polynom-Registeranschluss 46 anliegt, im sechsten Zellgatter 93 AND-verknüpft. Das Ausgangssignal des zweiten Erweiterungsadders 98 wird mit der Bitstelle  $PR_{n-1}$  des gespeicherten irreduziblen Polynoms PR 9, welche über den ersten Irreduzibel-Polynom-Registeranschluss 45 anliegt, im neunten Zellgatter 97 AND-verknüpft. Mit dessen Ausgangssignal und dem Ausgangssignal des sechsten Zellgatters 93 erfolgt im zweiten Ausgangsadder 100 eine Modulo-Addition.

Eine modulo-Addition erfolgt ebenfalls mit der Signalbelegung des dritten Zwischenergebnisanschlusses 41 und der Signalbelegung des siebenten Reduzier-Erweiterungs-Anschlusses 87 im ersten Ausgangsadder 99. Dessen Ausgangssignal wird mit dem Ausgangssignal des zweiten Ausgangsadders 100 im dritten Ausgangsadder 102 modulo-addiert und bildet mit seinem Ausgangssignal die Signalbelegung des Ergebnisregisteranschlusses  $E_{n-1}$  47.

Das Ausgangssignal des zweiten Erweiterungsadders 98 wird mit der Bitstelle  $PR_{n-2}$  des gespeicherten irreduziblen Polynoms PR 9, welche über den zweiten Irreduzibel-Polynom-Registeranschluss 46 anliegt, im zehnten Zellgatter 101 AND-verknüpft. Dessen Ausgangssignal wird mit der Signalbelegung des vierten Zwischenergebnisanschlusses 42 im vierten Ausgangsadder 103 modulo-addiert. Dieses Ausgangssignal wird mit der Signalbelegung des achten Reduzier-Erweiterungs-Anschluss 88 im fünften Ausgangsadder 104 modulo-addiert und bildet mit seinem Ausgangssignal die Signalbelegung des Ergebnisregisteranschlusses  $E_{n-2}$  48.

Wie in Fig. 8 ersichtlich, realisiert die Einstufen-Adder-Schlüsselkonfiguration 112 die Signalverarbeitung in zwei



vertikalen Ebenen und in nur einer horizontal entfalteten Adder-Stufe. Es gelangen über den ersten bis vierten Zwischenergebnisanschluss 39 bis 42 die von der Einstufen-Adder-Erweiterungsstruktur 111 bereitgestellten Stellenwerte des Zwischenergebnisses Z in die Einstufen-Adder-Schlüsselkonfiguration 112.

Außerdem gelangen über die Anschlüsse erster PEM-Anschluß 65, zweiter PEM-Anschluss 66, dritter PEM-Anschluss 63 und vierter PEM-Anschluss 78 die Bitstellen  $PE_{n-2,0}$  und  $PE_{n-1,0}$  sowie  $PE_{n-2,1}$  und  $PE_{n-1,1}$  der gespeicherten Matrix PEM 11 der erweiterten Form PE 7 des irreduziblen Polynoms PR 9 in die Einstufen-Adder-Schlüsselkonfiguration 112.

Die über den ersten Zwischenergebnisanschluss 39 bereitgestellte Bitstelle  $Z_{n+1}$  des Zwischenergebnisses Z wird einerseits am Eingang des zweiten Zellgatters 68 mit der Bitstelle  $PE_{n-2,0}$  der Matrix PEM 11 und andererseits am Eingang des ersten Zellgatters 67 mit der Bitstelle  $PE_{n-2,1}$  der Matrix-PEM 11 AND-verknüpft.

Die über den zweiten Zwischenergebnisanschluss 40 bereitgestellte Bitstelle  $Z_n$  des Zwischenergebnisses Z wird einerseits am Eingang des vierten Zellgatters 70 mit der Bitstelle  $PE_{n-1,0}$  der Matrix PEM und andererseits am Eingang des dritten Zellgatters 69 mit der Bitstelle  $PE_{n-1,1}$  der Matrix-PEM 11 AND-verknüpft.

Eine modulo-Addition erfolgt mit der Signalbelegung des dritten Zwischenergebnisanschlusses 41 und dem Ausgangssignal des vierten Zellgatters 70 im zweiten XOR-Teiladder 72.

Mit der Signalbelegung des dritten Reduzierer-Erweiterungsanschlusses 64 und dem Ausgangssignal des zweiten Zellgatters 68 erfolgt eine modulo-Addition im ersten XOR-Teiladder 71. Dessen Ausgangssignal wird mit dem Ausgangssignal des zweiten XOR-Teiladder 72 im dritten XOR-Teiladder 73 ebenfalls modulo-addiert und bildet mit seinem Ausgangssignal die Signalbelegung

des Ergebnisregisteranschlusses  $E_{n-1}$  47. Eine weitere modulo-Addition erfolgt mit der Signalbelegung des vierten Zwischenergebnisanschlusses 42 und dem Ausgangssignal des dritten Zellgatters 69 im fünften XOR-Teiladder 75. Mit der Signalbelegung des vierten Reduzierer-Erweiterungsanschlusses 77 und dem Ausgangssignal des ersten Zellgatters 67 erfolgt eine modulo-Addition im vierten XOR-Teiladder 74. Dessen Ausgangssignal wird mit dem Ausgangssignal des fünften XOR-Teiladders 75 im sechsten XOR-Teiladder 76 ebenfalls modulo-addiert und bildet mit seinem Ausgangssignal die Signalbelegung des Ergebnisregisteranschlusses  $E_{n-2}$  48.

15

20

25

30

35

Systemonic AG  
01099 Dresden

## Bezugszeichenliste

- |    |    |   |
|----|----|---|
| 15 | 1  | erstes Eingabe-Register                             |
|    | 2  | zweites Eingabe-Register                            |
|    | 3  | Bitmultiplizierer                                   |
|    | 4  | Partiell-Produkt-Adder                              |
|    | 5  | Q-Reduktion-Adderteil                               |
| 20 | 6  | Final-Adderteil                                     |
|    | 7  | erweiterte Form PE                                  |
|    | 9  | irreduzibles Polynom PR                             |
|    | 10 | Ergebnisregister                                    |
|    | 11 | Matrix PEM  |
| 25 | 12 | Preset-Register-Bank                                |
|    | 13 | Feldgrößenadaptionslogik                            |
|    | 14 | Decoder   |
|    | 16 | Partiell-Produktadder-Erweiterungsstruktur          |
|    | 17 | Reduzierer-Modulo-Erweiterungsstruktur              |
| 30 | 18 | Reduzierer-Modulo-Schlüsselkonfiguration            |
|    | 19 | Partiell-Produktadder-Schlüsselkonfiguration        |
|    | 20 | Galois-MUL  |
|    | 21 | erster Partiell-Produktadder-Erweiterungsanschluss  |
|    | 22 | zweiter Partiell-Produktadder-Erweiterungsanschluss |
| 35 | 23 | Bitstellenanschluss $a_{n-1}$                       |
|    | 24 | Bitstellenanschluss $a_{n-2}$                       |
|    | 25 | Bitstellenanschluss $b_{n-1}$                       |
|    | 26 | Bitstellenanschluss $b_{n-2}$                       |

	27	erstes XOR-Adderglied
	28	erstes XOR-Bitmultipliziererglied
	29	zweites XOR-Adderglied
	30	zweites XOR-Bitmultipliziererglied
5	31	drittes XOR-Adderglied
	32	drittes XOR-Bitmultipliziererglied
	33	viertes XOR-Adderglied
	34	viertes XOR-Bitmultipliziererglied
	35	fünftes XOR-Adderglied
10	36	fünftes XOR-Bitmultipliziererglied
	37	sechstes XOR-Adderglied
	38	sechstes Bitmultipliziererglied
	39	erster Zwischenergebnisanschluss
	40	zweiter Zwischenergebnisanschluss
15	41	dritter Zwischenergebnisanschluss
	42	vierter Zwischenergebnisanschluss
	43	dritter Partiell-Produktadder-Erweiterungsanschluss
	44	vierter Partiell-Produktadder-Erweiterungsanschluss
	45	erster Irreduzibel-Polynom-Registeranschluss
20	46	zweiter Irreduzibel-Polynom-Registeranschluss
	47	Ergebnisregisteranschluss $E_{n-1}$
	48	Ergebnisregisteranschluss $E_{n-2}$
	49	erster Reduzierer-Zwischenergebnisanschluss
	50	zweiter Reduzierer-Zwischenergebnisanschluss
25	51	dritter Reduzierer-Zwischenergebnisanschluss
	52	vierter Reduzierer-Zwischenergebnisanschluss
	53	erstes AND-Gatter
	54	zweites AND-Gatter
	55	erster XOR-Verknüpfen
30	56	zweiter XOR-Verknüpfen
	57	drittes AND-Gatter
	58	viertes AND-Gatter
	59	dritter XOR-Verknüpfen
	60	vierter XOR-Verknüpfen
35	61	erster Reduzierer-Erweiterungsanschluss
	62	zweiter Reduzierer-Erweiterungsanschluss
	63	dritter PEM-Anschluss
	64	dritter Reduzierer-Erweiterungsanschluss

	65	erster PEM-Anschluss
	66	zweiter PEM-Anschluss
	67	erstes Zellgatter
	68	zweites Zellgatter
5	69	drittes Zellgatter
	70	viertes Zellgatter
	71	erster XOR-Teiladder
	72	zweiter XOR-Teiladder
	73	dritter XOR-Teiladder
10	74	vierter XOR-Teiladder
	75	fünfter XOR-Teiladder
	76	sechster XOR-Teiladder
	77	vierter Reduzierer-Erweiterungs-Anschluss
	78	vierter PEM-Anschluss
15	81	fünfter Reduzierer-Erweiterungs-Anschluss
	82	erster Preset-Registeranschluss
	84	sechster Reduzier-Erweiterung-Anschluss
	85	zweiter Preset-Registeranschluss
	86	neunter Reduzier-Erweiterungs-Anschluss
20	87	siebter Reduzier-Erweiterung-Anschluss
	88	achter Reduzier-Erweiterung-Anschluss
	89	zehnter Reduzier-Erweiterung-Anschluss
	91	fünftes Zellgatter
	92	erster Erweiterungsadder
25	93	sechstes Zellgatter
	94	siebentes Zellgatter
	95	achtes Zellgatter
	96	erster Erweiterungsadder
	97	neuntes Zellgatter
30	98	zweiter Erweiterungsadder
	99	erster Ausgangsadder
	100	zweiter Ausgangsadder
	101	zehntes Zellgatter
	102	dritter Ausgangsadder
35	103	vierter Ausgangsadder
	104	fünfter Ausgangsadder
	105	Addiererteil
	106	Reduziererteil

- 107 Reduktions-Modulo-Anordnung
- 108 Zweistufen-Adder-Erweiterungsstruktur
- 109 Zweistufen-Adder-Schlüsselkonfiguration
- 110 Reduktions-Zweistufen-Adder-Anordnung
- 5 111 Einstufen-Adder-Erweiterungsstruktur
- 112 Einstufen-Adder-Schlüsselkonfiguration
- 113 Reduktions-Einstufen-Adder-Anordnung

10

15

**LIPPERT, STACHOW, SCHMIDT & PARTNER**

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49(0)351.3 18 18-0

Telefax +49(0)351.3 18 18 33

Ad-wb/wb

8. Februar 2001

5

Systemonic AG

01099 Dresden

10

# Verfahren und Anordnung zur Finiten Feld Multiplikation (FFM)

## Patentansprüche

15

1. Verfahren zur Ausführung einer Finiten Feld Multiplikation eines in einem ersten Eingangsregister eines mit den Bitstellen  $a_0 \dots a_{n-1}$  bereitgestellten ersten Galois-elementes  $a$ , welches mit einem zweiten, in einem zweiten Eingangsregister mit den Bitstellen  $b_0 \dots b_{n-1}$  bereitgestellten Galois-elementes  $b$  in einem digitalen Galois-Multiplizierer (MUL) multipliziert wird, wobei die Galoiselemente  $a$  und  $b$  zu einem Galoisfeld  $GF\ 2^n$ , welches durch ein irreduzibles Polynom  $PR$  mit den Bitstellen  $PR_0 \dots PR_{n-1}$  beschrieben wird, gehören, d a d u r c h g e k e n n z e i c h n e t, dass in einem Addierteil (105) des Galois-MUL (20), in einem Verarbeitungsschritt ein Zwischenergebnis  $Z$  von Zwischensummen partieller Produkte der Bitbreite  $2n-2$ , welches kein Element des durch das irreduzible Polynom  $PR$  (9) beschriebenen Galoisfeldes repräsentiert, gebildet wird, und dass in einem Reduzierteil (106) des Galois-MUL (20) das Zwischenergebnis  $Z$  mit den Bitstellen  $Z_{2n-2} \dots Z_0$  mittels einer Modulo-Division durch das irreduzible Polynom  $PR$  (9) mit den Bitstellen  $PR_{n-1} \dots PR_0$  verarbeitet wird und damit nach Durchlaufen aller XOR-Verküpfungen das Ergebnis  $E$  mit den Bitstellen  $E_{n-1} \dots E_0$  berechnet wird.

20

25

30

35

2. Verfahren nach Anspruch 1 d a d u r c h g e k e n n z e i c h n e t, dass in einem Reduzierteil (106) die

Modulo-Division des Zwischenergebnisses  $Z$  durch das irreduzible Polynom  $PR$  (9) zweistufig ausgeführt wird, indem in einer ersten Verarbeitungsstufe alle Bitsstellen  $Z_{2n-2} \dots Z_n$  mit der erweiterten Form  $PE$  (7) des irreduziblen Polynoms  $PR$  (9) mit den Bitsstellen  $PE_{n-1} \dots PE_0$  jeweils AND-verknüpft werden und danach mittels einer ersten parallel operierenden Adder-Baumstruktur, welche die logische Operation XOR ausführend realisiert, zusammengefaßt werden und diese zusammengefaßten Teilergebnisse nachfolgend in einer zweiten Verarbeitungsstufe mit den Bitsstellen  $PR_{n-1} \dots PR_0$  des irreduzible Polynom  $PR$  (9) jeweils AND-verknüpft und mittels einer zweiten parallel operierenden Adder-Baumstruktur, welcher auch die logische Operation XOR mehrfach ausführend realisiert und jeweils zusammen mit den Bitsstellen  $Z_{n-1} \dots Z_0$  des Zwischenergebnisses  $Z$  zum Ergebnis  $E$  mit den Bitsstellen  $E_{n-1} \dots E_0$  zusammenfaßt werden.

3. Verfahren nach Anspruch 1 d a d u r c h g e k e n n - z e i c h n e t, dass eine Matrix  $PEM$  (11) der Bitsstellenanordnung

$$PEM = \begin{pmatrix} PE_{n-1,n-2} & \dots & PE_{n-1,0} \\ \vdots & & \vdots \\ PE_{0,n-2} & \dots & PE_{0,0} \end{pmatrix}$$

bezüglich der erweiterten Form  $PE$  (7) des irreduziblen Polynoms  $PR$  (9) vorausberechnet wird, dass die Modulo-Division des Zwischenergebnisses  $Z$  durch das irreduzible Polynom  $PR$  (9) dadurch ausgeführt wird, dass alle Bits  $Z_{2n-2} \dots Z_n$  mit der Matrix  $PEM$  (11) der erweiterten Form  $PE$  (7) des irreduziblen Polynoms  $PR$  (9) mit den Bitsstellen  $PE_{n-1,n-2} \dots PE_{n-1,0} \dots PE_{0,n-2} \dots PE_{0,0}$  jeweils AND-verknüpft und danach mittels einer dritten parallel operierenden Adder-Baumstruktur, welche die logische Operation XOR mehrfach ausführend realisiert, jeweils zusammen mit den Bitsstellen  $Z_{n-1} \dots Z_0$  des Zwischenergebnisses  $Z$  zum Ergebnis  $E$  mit den Bitsstellen  $E_{n-1} \dots E_0$  zusammenfaßt wird.



4. Verfahren nach Anspruch 2 und 3 d a d u r c h g e -  
 k e n n z e i c h n e t, dass bei variablen Galoiselemen-  
 ten  $v_a < a$  und  $v_b < b$ , mit einer hierbei resultierenden Bit-  
 breite  $2m-2$  des Zwischenergebnisses  $Z$  zur Anpassung an  
 unterschiedliche Galois-Felder, alle Bitstellen des Zwi-  
 schenergebnisses  $Z$  um  $Z_{2m_{max}-2} - Z_m$  Stellen mittels eines Deco-  
 ders (14) und einer Feldgrößenadaptionslogik (13) ver-  
 schoben werden, bevor sie mit der erweiterten Form PE (7)  
 des irreduziblen Polynoms PR (9) verknüpft werden, wobei  
 die Bitbreite  $2m_{max}$  die maximale Bitbreite des Zwischen-  
 ergebnisses  $Z$  bei jeweils maximalen Bitbreite  $n$  der Ga-  
 loiselemente  $a$  und  $b$  darstellt.
5. Anordnung zur Finiten Feld Multiplikation in einem Ga-  
 lois-Multiplizierer (MUL), der aus einem Zellverbund ein-  
 zelner zellularer Array-Multiplizierer besteht, welche zu  
 einer Reduktions-Modulo-Anordnung zusammengeschaltet sind,  
 die die Struktur eines MSB-First Matrix-Galois-Multipli-  
 zierers aufweist, d a d u r c h g e k e n n z e i c h -  
 n e t, dass der Galois-MUL (20) einerseits aus einem Ad-  
 dierer-Teil (105), der wiederum aus einer Partiell-Pro-  
 duktadder-Erweiterungsstruktur (16) und aus einer  
 Partiell-Produktadder-Schlüssel-Konfiguration (19) zu-  
 sammengesetzt ist, wobei diese miteinander durch einen  
 ersten Partiell-Produktadder-Erweiterungsanschluss (22),  
 einen zweiten Partiell-Produktadder-Erweiterungsanschluss  
 (23), einen dritten Partiell-Produktadder-Erweiterungs-  
 anschluss (43) und einen vierten Partiell-Produktadder-  
 Erweiterungsanschluss (43) verbunden sind, besteht, wobei  
 weiterhin der Addierer-Teil (105) mit einem ersten Ein-  
 gaberegister (1) und einem zweiten Eingaberegister (2)  
 verbunden ist und andererseits der Galois-MUL (20) aus einem  
 Reduzierer-Teil (106), welcher mit einem Zwischenergebnis-  
 ausgang des Partiellen-Produktadders (4), mit einer er-  
 weiterte Form (7) und/oder mit einem irreduziblen Polynom  
 PR (9) oder mit einer die Matrix PEM (11) enthaltenden  
 Preset-Register-Bank (12) und einem Ergebnisregister (10)  
 verbunden ist, besteht.

6. Anordnung nach Anspruch 5 d a d u r c h g e k e n n -  
 z e i c h n e t, dass in der Partiell-Produktadder-Schlüs-  
 sel-Konfiguration (19) ein erster Eingang eines ersten  
 XOR-Adderglied (27) mit einem ersten Partiell-  
 Produktadder-Erweiterungsanschluss (21) verbunden ist,  
 dass das erste XOR-Adderglied (27) mit einem zweiten Ein-  
 gang mit einem Ausgang eines ersten XOR-Bitmultiplizierers  
 (28) verbunden ist, dass ein Ausgang des ersten XOR-Adder-  
 gliedes (27) mit einem ersten Zwischenergebnisanschluss  
 (39) verbunden ist, dass der erste XOR-Bitmultiplizierer  
 (28) mit einem ersten Eingang an einen Bitstellenanschluss  
 $a_{n-1}$  (23) angeschlossen ist, dass ein zweiter Eingang des  
 ersten XOR-Bitstellenmultiplizierers (28) mit einem Bit-  
 stellenanschluss  $b_{n-1}$  (25) geschaltet ist, dass ein erster  
 Eingang eines zweiten XOR-Adderglied (29) mit einem zwei-  
 ten Partiell-Produktadder-Erweiterungsanschluss (22) ver-  
 bunden ist, dass ein zweiter Eingang des zweiten XOR-Ad-  
 dergliedes (29) mit einem Ausgang eines dritten XOR-Adder-  
 gliedes (31) verbunden ist, dass ein Ausgang des zweiten  
 XOR-Addergliedes (29) an einen zweiten Zwischenergebnis-  
 anschluss (40) geschaltet ist, dass ein erster Eingang des  
 dritten XOR-Addergliedes (31) mit einem Ausgang eines  
 zweiten XOR-Bitmultiplizierergliedes (30) geschaltet ist,  
 dass ein zweiter Eingang des dritten XOR-Addergliedes (31)  
 mit einem Ausgang eines dritten XOR-Bitmultiplizierer-  
 gliedes (32) geschaltet ist, dass ein erster Eingang des  
 zweiten XOR-Bitmultiplizierergliedes (30) an den Bitstel-  
 lenanschluss  $a_{n-1}$  (23) angeschlossen ist, dass ein zweiter  
 Eingang des zweiten XOR-Bitmultiplizierergliedes (30) an  
 einen Bitstellenanschluss  $b_{n-2}$  (26) geschaltet ist, dass  
 ein erster Eingang des dritten XOR-Bitmultipliziererglie-  
 des (32) mit einem Bitstellenanschluss  $a_{n-2}$  (24) verbunden  
 ist, dass ein zweiter Eingang des dritten XOR-Bitmulti-  
 plizierergliedes (32) mit einem Bitstellenanschluss  $b_{n-1}$  (25)  
 geschaltet ist, dass ein Ausgang eines vierten XOR-Adder-  
 gliedes (33) mit einem ersten Eingang eines fünften XOR-  
 Addergliedes (35) verbunden ist, dass ein Ausgang des  
 fünften XOR-Addergliedes (35) mit einem dritten Zwischen-

ergebnisanschluss (41) verbunden ist, dass ein Eingang des  
 vierten XOR-Addergliedes (33) mit einem Ausgang eines  
 vierten XOR-Bitmultiplicierergliedes (34) verbunden ist,  
 dass ein erster Eingang des vierten XOR-Bitmultiplicierergliedes (34) mit dem Bitstellenanschluss  $a_{n-1}$  (23) verbunden ist, dass ein zweiter Eingang des vierten XOR-Bitmultiplicierergliedes (34) an einen Bitstellenanschluss  $b_{n-2}$  (26) geschaltet ist, dass ein zweiter Eingang des vierten XOR-Addergliedes (33) mit einem Ausgang eines fünften XOR-Bitmultiplicierergliedes (36) geschaltet ist, dass ein erster Eingang eines fünften XOR-Bitmultiplicierergliedes (36) mit dem Bitstellenanschluss  $a_{n-2}$  (24) geschaltet ist, dass ein erster Eingang des fünften XOR-Bitmultiplicierergliedes (36) mit dem Bitstellenanschluss  $a_{n-2}$  (24) geschaltet ist, dass ein zweiter Eingang des fünften XOR-Bitmultiplicierergliedes (36) mit dem Bitstellenanschluss  $b_{n-1}$  (25) geschaltet ist, dass ein zweiter Eingang des fünften XOR-Addergliedes (35) mit einem vierten partiell-Produktadder-Erweiterungsanschluss (44) verbunden ist, dass ein Ausgang eines sechsten XOR-Addergliedes (37) mit einem vierten Zwischenergebnisanschluss (42) verbunden ist, dass ein erster Eingang des sechsten XOR-Addergliedes (37) mit einem Ausgang eines sechsten XOR-Bitmultiplicierergliedes (38) geschaltet ist, welches wiederum mit seinem ersten Eingang an den Bitstellenanschluss  $a_{n-2}$  (24) geschaltet ist, dass ein zweiter Eingang des sechsten XOR-Bitmultiplicierergliedes (38) an den Bitstellenanschluss  $b_{n-2}$  (26) geschaltet ist, dass ein zweiter Eingang des sechsten XOR-Addergliedes (37) mit einem dritten Partiell-Produktadder-Erweiterungsanschluss (43) geschaltet ist.

7. Anordnung nach Anspruch 5 d a d u r c h g e k e n n -  
 z e i c h n e t, dass der Reduzierteil (106) aus einer  
 Reduktions-Modulo-Anordnung besteht (107), welche eine  
 Reduzierer-Modulo-Erweiterungsstruktur (17) und eine Redu-  
 zierer-Modulo-Schlüsselkonfiguration (19) enthält.

8. Anordnung nach Anspruch 7 d a d u r c h g e k e n n -  
 z e i c h n e t, dass in der Reduzierer-Modulo-Schlüssel-  
 konfiguration (19) ein erster Eingang eines ersten AND-  
 Gatters (53) einerseits mit einem ersten Reduzierer-Zwi-  
 schenergebnisanschluss (49) und andererseits mit einem  
 5 zweiten Reduzierer-Erweiterungsanschluss (62) verbunden  
 ist, dass ein zweiter Eingang des ersten AND-Gatters (53)  
 einerseits mit einem ersten Irreduzibel-Polynom-Regist-  
 ersanschluss (45) und andererseits mit einem ersten Eingang  
 10 eines dritten AND-Gatters (57) geschaltet ist, dass ein  
 Ausgang des ersten AND-Gatters (53) mit einem ersten Ein-  
 gang eines ersten XOR-Verknüpfers (55) verbunden ist, dass  
 ein zweiter Eingang des ersten XOR-Verknüpfers (55) mit  
 einem zweiten Reduzierer-Zwischenergebnisanschluss (50)  
 15 geschaltet ist, dass ein Ausgang des ersten XOR-Verknüp-  
 fers (55) einerseits mit einem ersten Eingang eines drit-  
 ten AND-Gatters (57) und weiterhin mit einem ersten Redu-  
 zierer-Erweiterungsanschluss (61) und außerdem mit einem  
 ersten Eingang eines vierten AND-Gatters (58) verbunden  
 20 ist, dass ein Ausgang des dritten AND-Gatters (57) mit  
 einem ersten Eingang eines dritten XOR-Verknüpfers (59)  
 geschaltet ist, dass ein Ausgang des dritten XOR-Verknüp-  
 fers (59) mit einem Ergebnisregisteranschluss  $E_{n-1}$  (47)  
 verbunden ist, dass ein zweiter Eingang des zweiten AND-  
 25 Gatters (54) einerseits mit einem zweiten Irreduzibel-  
 Polynom-Registeranschluss (46) und außerdem mit einem  
 zweiten Eingang des vierten AND-Gatters (58) verbunden  
 ist, dass ein Ausgang des vierten AND-Gatters (58) mit  
 einem ersten Eingang eines vierten XOR-Verknüpfers (60)  
 30 geschaltet ist, dass ein Ausgang des vierten XOR-Verknüp-  
 fers (60) mit einem Ergebnisanschluss  $E_{n-2}$  (48) verbunden  
 ist, dass ein Ausgang des zweiten AND-Gatters (54) mit  
 einem ersten Eingang eines zweiten XOR-Verknüpfers (56)  
 geschaltet ist, dass ein zweiter Eingang des zweiten XOR-  
 35 Verknüpfers (56) mit einem Reduzierer-Zwischenergebnis-  
 anschluss (51) verbunden ist, dass ein Ausgang des zweiten  
 XOR-Verknüpfers (56) mit einem zweiten Eingang des dritten  
 XOR-Verknüpfers (59) verbunden ist, dass ein zweiter Ein-

gang des vierten XOR-Verknüpfers (60) mit einem vierten Reduzierer-Zwischenergebnisanschluss (52) geschaltet ist.

9. Anordnung nach Anspruch 5 d a d u r c h g e k e n n -  
z e i c h n e t, dass der Reduzierteil (106) aus einer Reduktions-Zweistufen-Adder-Anordnung (110) besteht, welche eine Zweistufen-Adder-Erweiterungsstruktur (108) und eine Zweistufen-Adder-Schlüsselkonfiguration (109) enthält.

10. Anordnung nach Anspruch 12 d a d u r c h g e k e n n -  
z e i c h n e t, dass in der Zweistufen-Adder-Schlüsselkonfiguration (109) ein erster Eingang eines fünften Zell-Gatters (91) einerseits mit einem ersten Eingang eines siebenten Zell-Gatters (94) und außerdem mit dem ersten Zwischenenergebnisanschluss (39) geschaltet ist, dass ein zweiter Eingang des fünften Zell-Gatters (91) einerseits mit einem zweiten Preset-Registeranschluss (85) und andererseits mit einem ersten Eingang eines achten Zell-Gatters (95) verbunden ist, dass ein zweiter Eingang des achten Zell-Gatters (95) mit dem zweiten Zwischenenergebnisanschluss (40) geschaltet ist, dass ein Ausgang des achten Zellgatters (95) mit einem ersten Eingang eines ersten Erweiterungsadders (96) geschaltet ist, dass ein Ausgang des siebenten Zellgatters (94) mit einem zweiten Eingang des ersten Erweiterungsadders (96) geschaltet ist, dass ein zweiter Eingang des siebenten Zellgatters (94) mit einem ersten Preset-Registeranschluss (82) verbunden ist, dass ein Ausgang des fünften Zellgatters (91) mit einem ersten Eingang eines ersten Erweiterungsadders (92) verbunden ist, dass ein zweiter Eingang des ersten Erweiterungsadders (92) mit einem sechsten Reduzier-Erweiterungs-Anschluss (84) verbunden ist, dass ein Ausgang des ersten Erweiterungsadders (92) einerseits mit einem ersten Eingang eines sechsten Zellgatters (93) und anderseit mit einem neunten Reduzier-Erweiterungs-Anschluss (86) geschaltet ist, dass ein zweiter Eingang des sechsten Zellgatters (93) einerseits mit dem zweiten Irreduzibel-Poly-

nom-Registeranschluss (46) und andererseits mit einem  
 ersten Eingang eines zehnten Zellgatters (101) verbunden  
 ist, dass ein Ausgang des ersten Erweiterungsadders (96)  
 mit einem ersten Eingang eines zweiten Erweiterungsadders  
 5 (98) verbunden ist, dass ein zweiter Eingang des zweiten  
 Erweiterungsadders (98) mit einem fünften Reduzierer-Er-  
 weiterungs-Anschluss (81) verbunden ist, dass ein erster  
 Eingang eines neunten Zellgatters (97) mit dem ersten  
 Irreduzibel-Polynom-Registeranschluss (45) verbunden ist,  
 10 dass ein zweiter Eingang des neunten Zellgatters (97)  
 einerseits mit einem Ausgang des zweiten Erweiterungs-  
 adders (98) und andererseits mit einem zweiten Eingang des  
 zehnten Zellgatters (101) und außerdem mit einem zehnten  
 Reduzier-Erweiterungs-Anschluss (89) geschalten ist, dass  
 15 ein Ausgang des neunten Zellgatters (97) mit einem einem  
 ersten Eingang eines zweiten Ausgangsadders (100)  
 geschalten ist, dass ein Ausgang des sechsten Zellgatters  
 (93) mit einem zweiten Eingang des zweiten Ausgangsadders  
 (100) geschalten ist, dass ein Ausgang des zweiten Aus-  
 20 gangsadders (100) mit einem ersten Eingang eines dritten  
 Ausgangsadders (102) geschalten ist, dass ein erster Ein-  
 gang eines ersten Ausgangsadders (99) mit einem siebenten  
 Reduzier-Erweiterungs-Anschluss (87) geschalten ist, dass  
 ein zweiter Eingang des ersten Ausgangsadders (99) mit dem  
 25 dritten Zwischenergebnisanschluss (41) geschalten ist,  
 dass ein Ausgang des ersten Ausgangsadders (99) mit einem  
 zweiten Eingang des dritten Ausgangsadders (102) verbunden  
 ist, dass ein Ausgang des dritten Ausgangsadders (102) mit  
 dem Ergebnisanschluss  $E_{n-1}$  (47) verbunden ist, dass ein  
 30 Ausgang des zehnten Zellgatters (101) mit einem ersten  
 Eingang eines vierten Ausgangsadders (103) geschalten ist,  
 dass ein zweiter Eingang des vierten Ausgangsadders (103)  
 mit dem vierten Zwischenergebnisanschluss (42) geschalten  
 ist, dass ein Ausgang des vierten Ausgangsadders (103) mit  
 35 einem ersten Eingang eines fünften Ausgangsadders (104)  
 geschalten ist, dass ein zweiter Eingang des fünften Aus-  
 gangsadders (104) mit einem achten Reduzier-Erweiterungs-  
 Anschluss (88) verbunden ist, dass ein Ausgang des fünften

Ausgangsadders (104) mit dem Ergebnisanschluss  $E_{n-2}$  (48) verbunden ist.

- 5      11. Anordnung nach Anspruch 5 d a d u r c h g e k e n n -  
z e i c h n e t, dass der Reduzierteil (106) aus einer  
Reduktions-Einstufen-Adder-Anordnung (113) besteht, welche  
eine Einstufen-Adder-Erweiterungsstruktur (111) und einer  
Einstufen-Schlüsselkonfiguration (112) enthält.
- 10      12. Anordnung nach Anspruch 15 d a d u r c h g e k e n n -  
z e i c h n e t, dass in der Einstufen-Adder-Schlüssel-  
konfiguration (112) ein erster Eingang eines zweiten Zell-  
gatters (68) einerseits mit dem ersten Zwischenergebnis-  
anschluss (39) und andererseits mit einem ersten Eingang  
15      eines ersten Zellgatters (67) verbunden sind, dass ein  
zweiter Eingang des zweiten Zellgatters (68) mit einem  
ersten PEM-Anschluss (65) verbunden ist, dass ein Ausgang  
des zweiten Zellgatters (68) mit einem ersten Eingang  
eines ersten XOR-Teiladders (71) geschaltet ist, dass ein  
20      zweiter Eingang des ersten XOR-Teiladders (71) mit einem  
dritten Reduzierer-Erweiterungsanschluss (64) verbunden  
ist, dass ein zweiter Eingang des ersten Zellgatters (67)  
einem vierten PEM-Anschluss (78) geschaltet ist, dass ein  
erster Eingang eines vierten Zellgatters (70) einerseits  
25      mit dem zweiten Zwischenergebnisanschluss (40) und ande-  
rererseits mit einem ersten Eingang eines dritten Zellgat-  
ters (69) verbunden ist, dass ein zweiter Eingang des  
vierten Zellgatters (70) mit einem zweiten PEM-Anschluss  
(66) verbunden ist, dass ein Ausgang des vierten Zell-  
30      gatters (70) mit einem ersten Eingang eines zweiten XOR-  
Teiladders (72) geschaltet ist, dass ein zweiter Eingang  
des zweiten XOR-Teiladders (72) mit dem dritten Zwischen-  
ergebnisanschluss (41) geschaltet ist, dass ein Ausgang  
des zweiten XOR-Teiladders (72) mit einem ersten Eingang  
35      eines dritten XOR-Teiladders (73) geschaltet ist, dass ein  
Ausgang des ersten XOR-Teiladders (71) mit einem zweiten  
Eingang des dritten XOR-Teiladders (73) geschaltet ist,  
dass ein Ausgang des dritten XOR-Teiladders (73) mit dem

Ergebnisanschluss  $E_{n-1}$  (47) geschaltet ist, dass ein Ausgang des dritten Zellgatters (69) mit einem Eingang eines fünften XOR-Teiladders (75) verbunden ist, dass ein zweiter Eingang des dritten Zellgatters mit einem dritten PEM-Anschluss (63) geschaltet ist, dass ein Ausgang des ersten Zellgatters (67) mit einem ersten Eingang eines vierten XOR-Teiladders (74) geschaltet ist, dass ein zweiter Eingang des vierten XOR-Teiladders (74) mit einem vierten Reduzierer-Erweiterungs-Anschluss (77) geschaltet ist, dass ein zweiter Eingang des fünften XOR-Teiladders (75) mit dem vierten Zwischenergebnisanschluss (42) geschaltet ist, dass ein Ausgang des fünften XOR-Teiladders (75) mit einem ersten Eingang eines sechsten XOR-Teiladders (76) geschaltet ist, dass ein Ausgang des vierten XOR-Teiladders (74) mit einem zweiten Eingang des sechsten XOR-Teiladders (76) verbunden ist und dass dessen Ausgang mit dem Ergebnisanschluss  $E_{n-2}$  (48) geschaltet ist.

13. Anordnung nach den Ansprüchen 5 bis 10, d a d u r c h g e k e n n z e i c h n e t, dass ein Galois-Multiplizierer-Accumulator (MAC) angeordnet ist, der den Galois-MUL (20), das erste Eingaberegister (1), das zweite Eingaberegister (2), die erweiterte Form (7) und/oder das irreduzible Polynom PR (9), das Ergebnisregister (10) und ein Addierer enthält, wobei der Ausgang des Galois-MUL (20) mit dem Eingang des Addierers und dessen Ausgang mit dem Eingang des Ergebnisregisters (10) und der Ausgang des Ergebnisregisters (10) mit dem ersten Eingaberegister (1) und/oder dem zweiten Eingaberegister (2) geschaltet ist.

14. Anordnung nach den Ansprüchen 5, 6 und 11, 12, d a d u r c h g e k e n n z e i c h n e t, dass ein Galois-Multiplizierer-Accumulator (MAC) angeordnet ist, der den Galois-MUL (20), das erste Eingaberegister (1), das zweite Eingaberegister (2), die die Matrix PEM (11) enthaltende Preset-Register-Bank (12), das Ergebnisregister (10) und einen Addierer enthält, wobei der Ausgang des Galois-MUL (20) mit dem Eingang des Addierers und dessen Ausgang mit



dem Eingang des Ergebnisregisters (10) und der Ausgang des Ergebnisregisters (10) mit einem Eingang des ersten Eingaberegisters (1) und/oder einem Eingang des zweiten Eingaberegisters (2) geschaltet ist.

5

10

8. Februar 2001

5

**Systemonic AG**

**01099 Dresden**

10

**Verfahren und Anordnung zu Finiten Feld Multiplikation  
(FFM)**

**Zusammenfassung**

Die Erfindung betrifft Verfahren und Anordnungen zur Ausführung einer Finiten Feld Multiplikation von Galoiselementen in einem digitalen Galois-Multiplizierer (MUL).

20

Die erfindungsgemäße Lösung beschreibt Verfahren und Anordnungen, bei denen eine beschleunigte Multiplikation durch Operationen in parallel geführten Adder-Baumstrukturen erzielt wird, wobei damit Varianten von Modulo-Operationen mit einem vorher berechneten Hilfspolynom bzw. einer Hilfspolynom-Matrix realisiert werden.

25

Weiterhin wird ein Beschleunigungsverfahren beschrieben, bei dem eine Anpassung des Multiplizierfeldes an die Größe der Operanden durchgeführt wird. (Fig. 1)

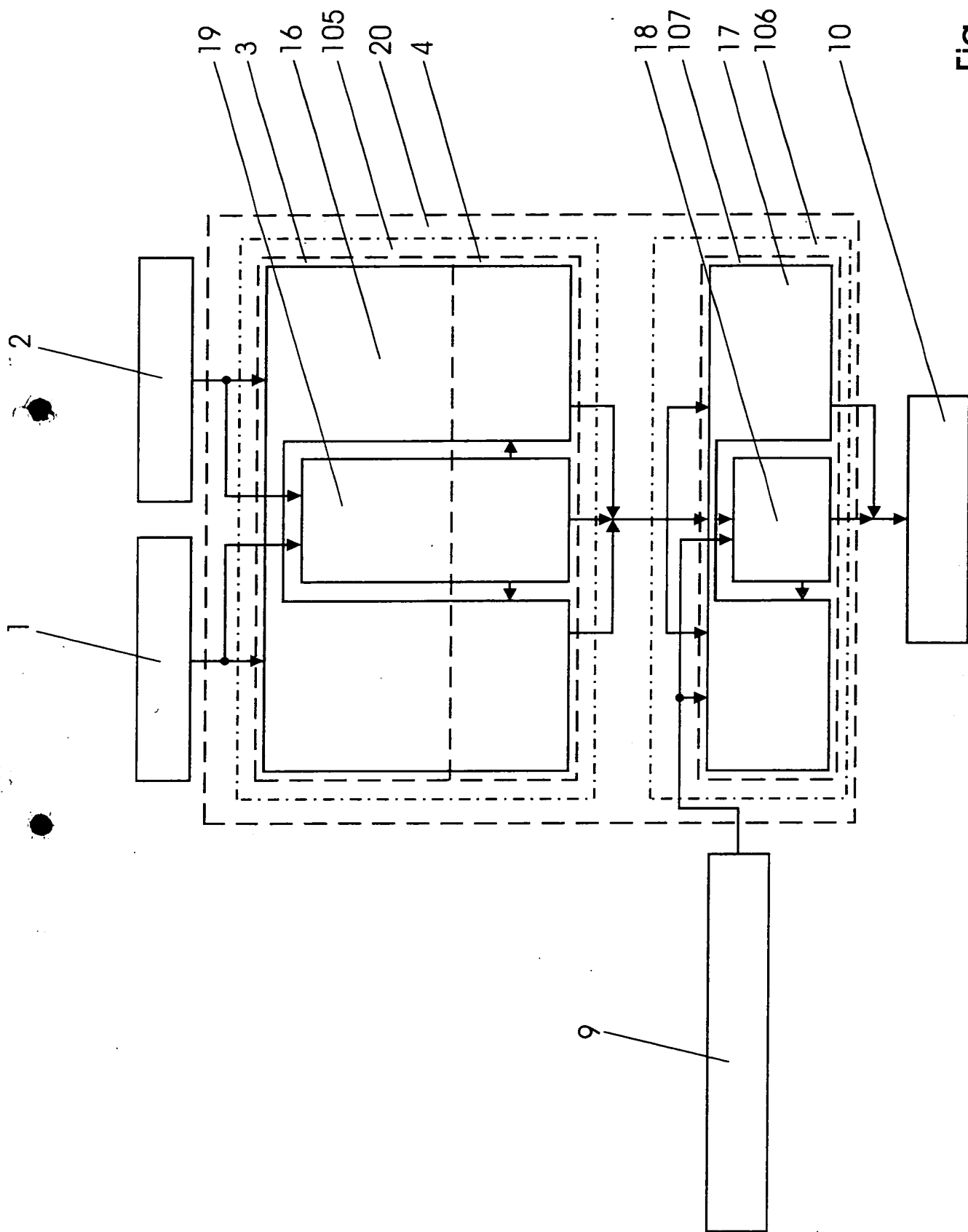


Fig. 1

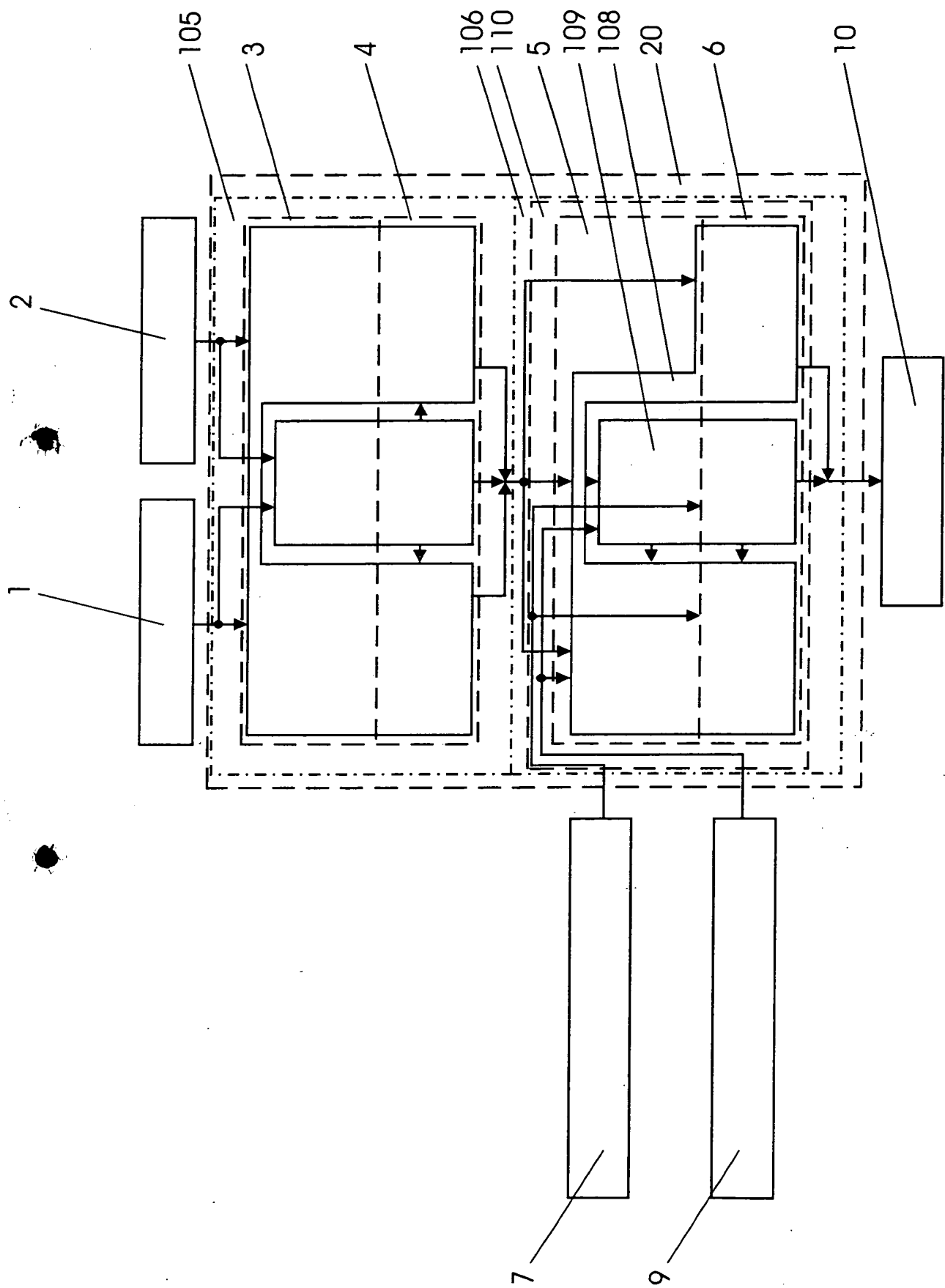


Fig. 2

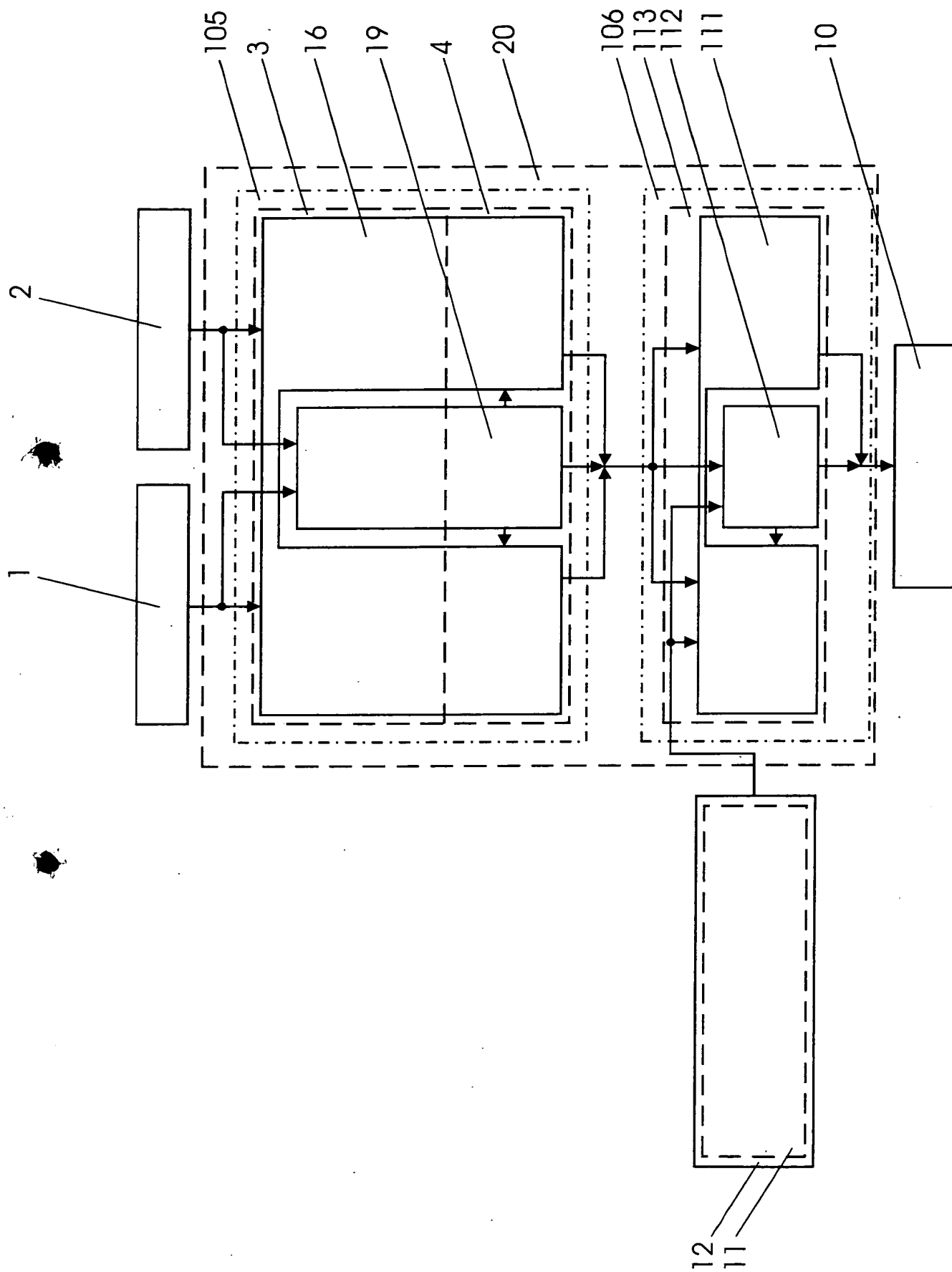


Fig. 3

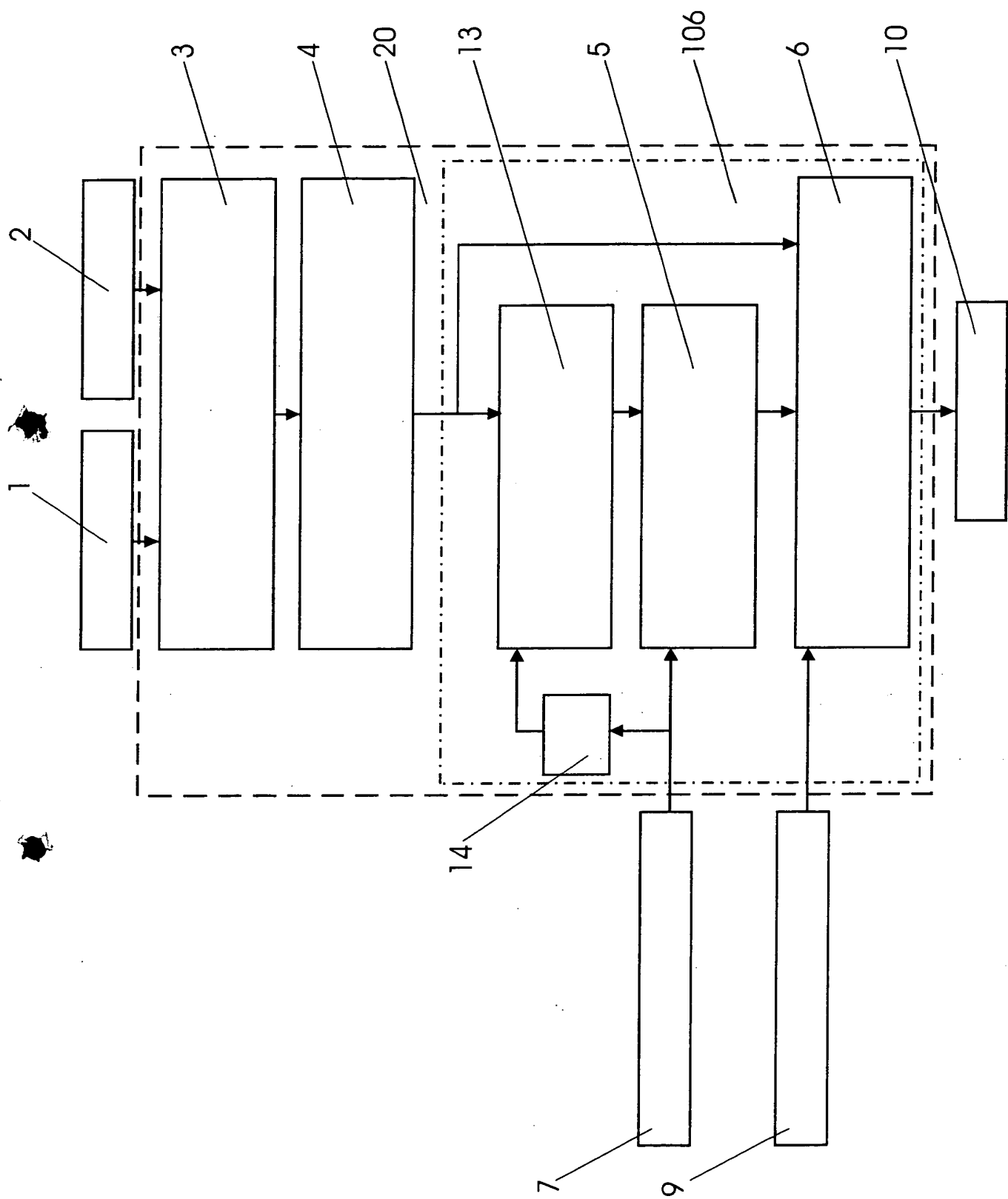


Fig. 4

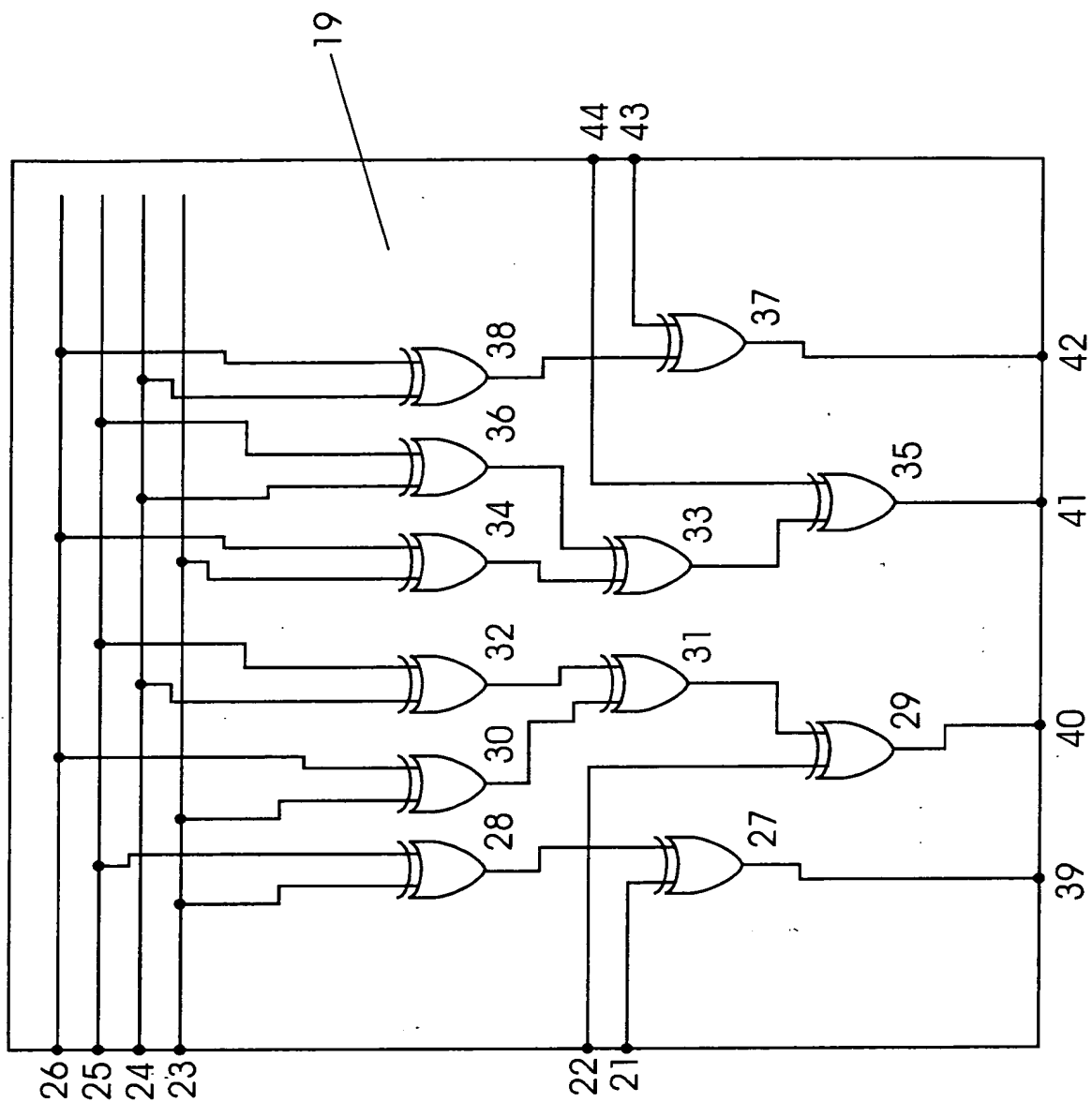


Fig. 5

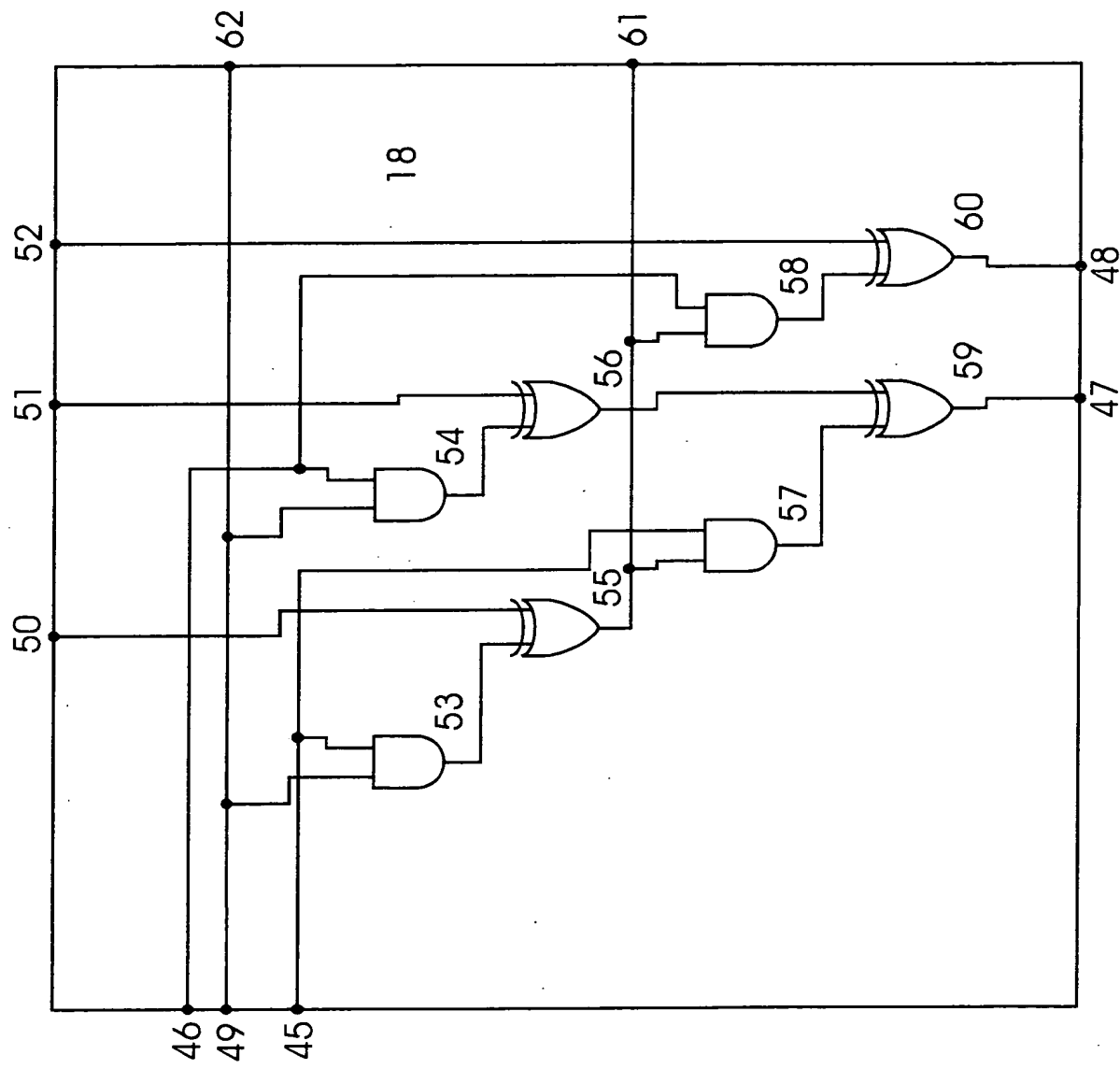


Fig. 6



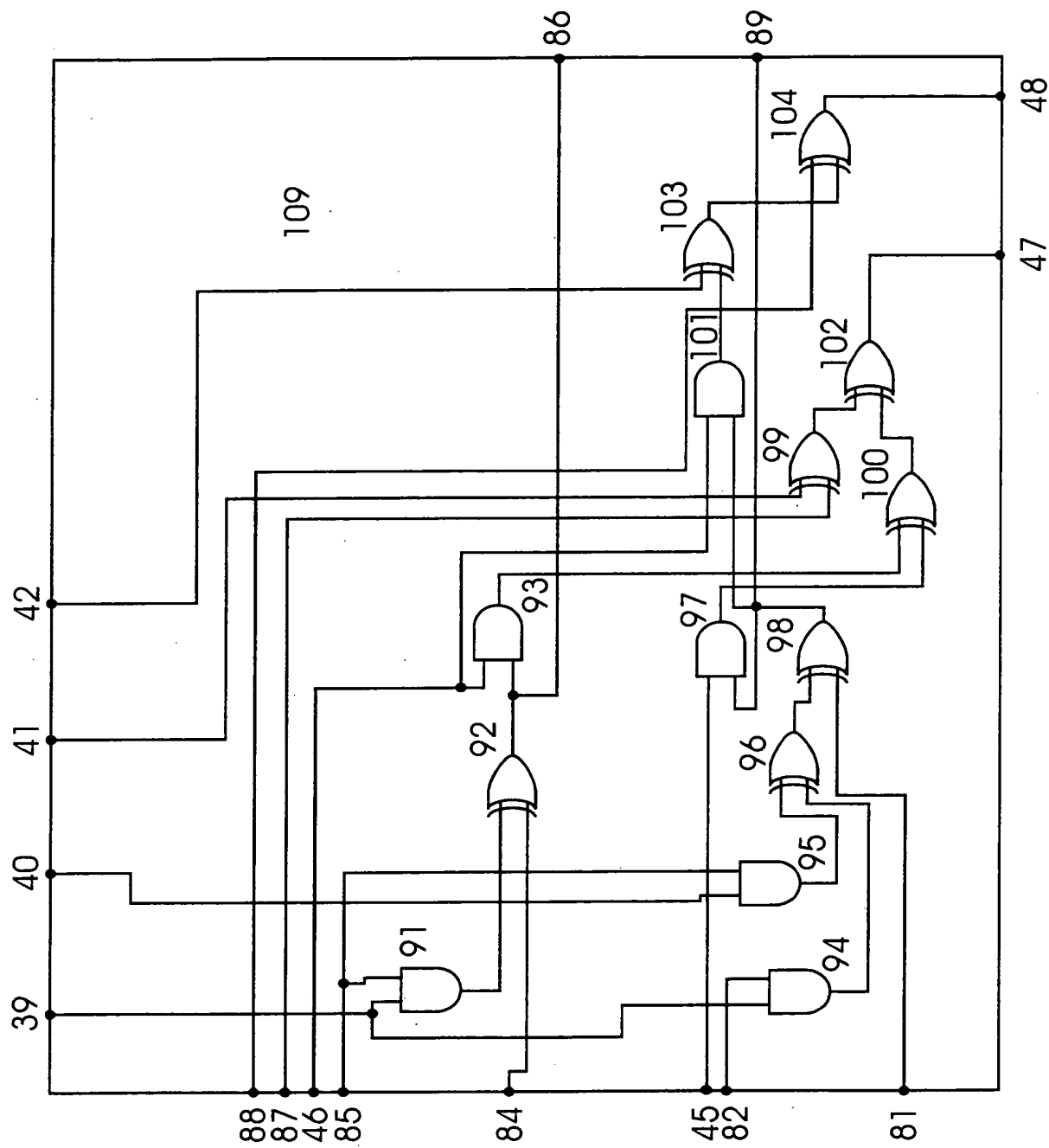


Fig. 7

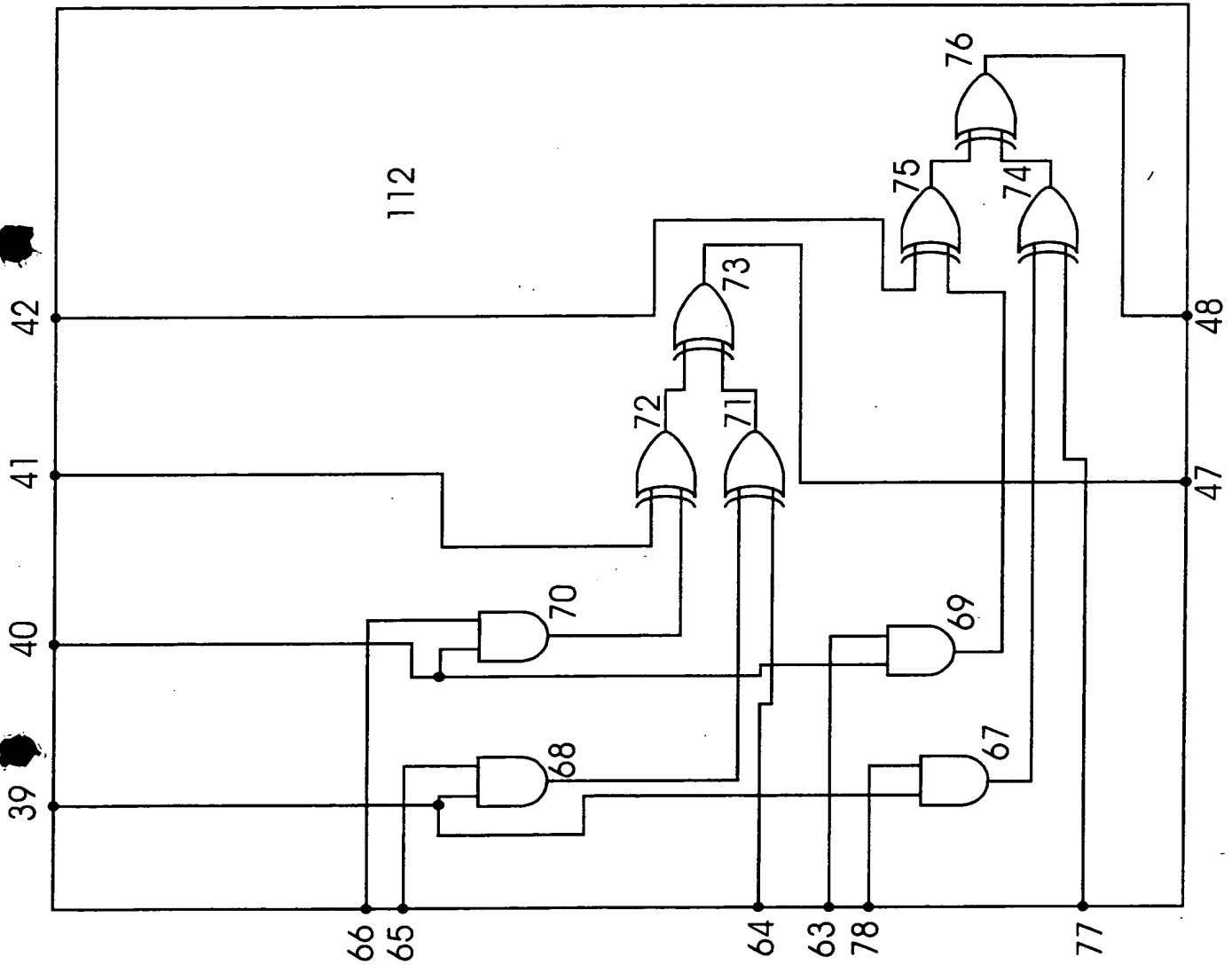


Fig. 8

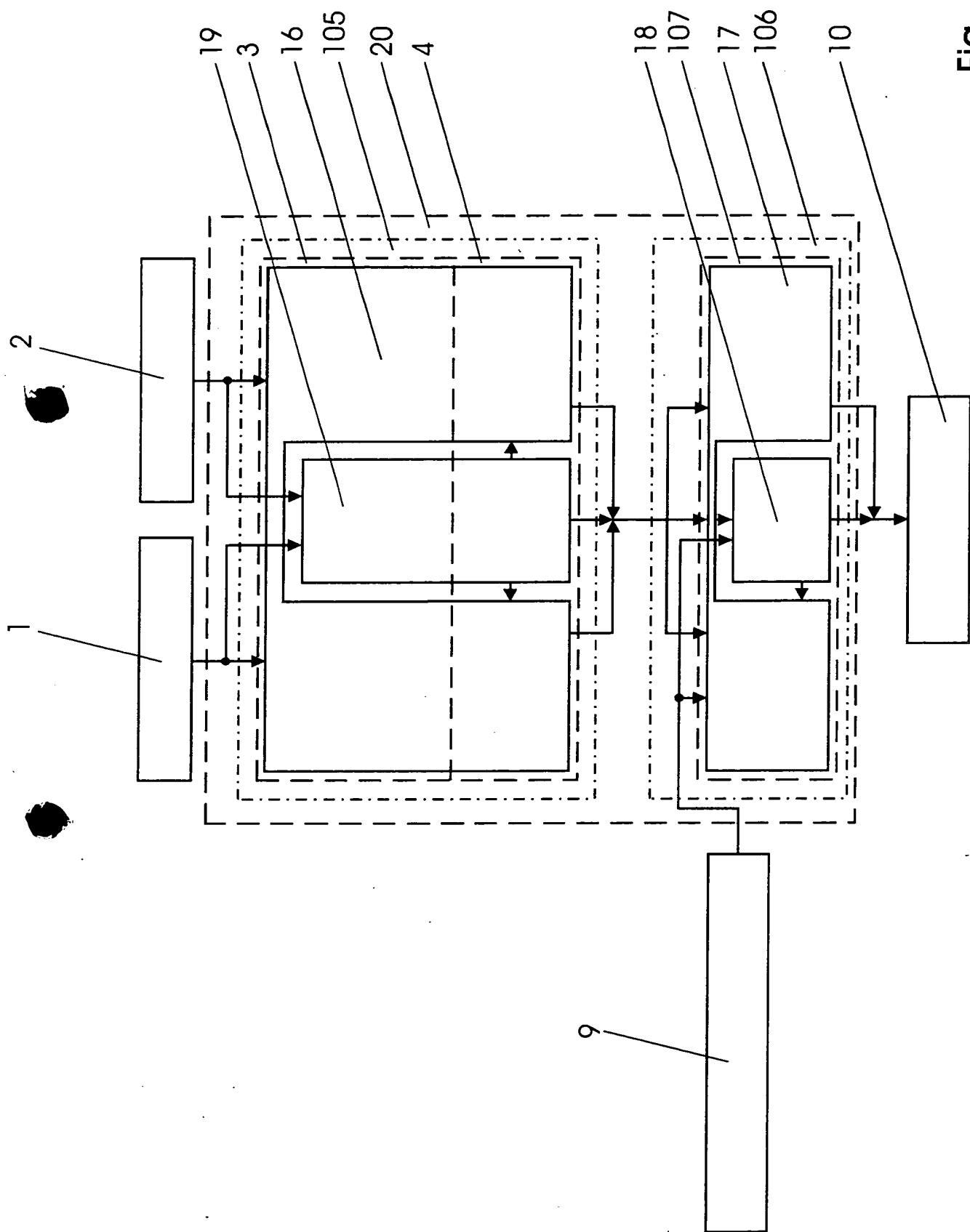


Fig. 1